

# Open Research Online

---

The Open University's repository of research publications and other research outputs

## Dealing with abstraction: case study generalisation as a method for eliciting design patterns

### Journal Item

#### How to cite:

Winters, Niall and Mor, Yishay (2009). Dealing with abstraction: case study generalisation as a method for eliciting design patterns. *Computers in Human Behavior*, 25(5) pp. 1079–1088.

For guidance on citations see [FAQs](#).

© 2009 Elsevier Ltd.



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Accepted Manuscript

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.1016/j.chb.2009.01.007>

---

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

---

[oro.open.ac.uk](http://oro.open.ac.uk)

# Dealing with abstraction: case study generalisation as a method for eliciting design patterns

Niall Winters and Yishay Mor

London Knowledge Lab

Institute of Education, University of London

23-29 Emerald Street

London WC1N 3QS

UK

{n.winters, [y.mor](mailto:y.mor@ioe.ac.uk)}@ioe.ac.uk

To appear in Symeon Retalis, Yannis Dimitriadis, Peter Goodyear (ed.) *Computers in Human Behavior*, special issue on *Design Patterns for Augmenting E-Learning Experiences*

## Keywords

Design patterns, generalisation, abstraction, case study

## Abstract

Developing a pattern language is a non-trivial problem. A critical requirement is a method to support pattern writers with abstraction, so as they can produce generalised patterns. In this paper, we address this issue by developing a *structured process of generalisation*. It is important that this process is initiated through engaging participants in identifying initial patterns, i.e. directly dealing with the 'cold-start' problem. We have found that short case study descriptions provide a productive 'way into' the process for participants. We reflect on a 1-year interdisciplinary pan-European research project involving the development of almost 30 cases and over 150 patterns. We provide example cases, detailing the process by which their associated patterns emerged. This was based on a foundation for generalisation from cases with common attributes. We discuss the merits of this approach and its implications for pattern development.

# 1. Introduction

Developing technologies and associated pedagogical resources that meet the needs of learners is a key challenge for the field of technology enhanced learning (TEL). This challenge presents a number of complexities for researchers. To take just one example, the innovator/practitioner gap has remained a significant stumbling block to interdisciplinary research. The simple truth is that many TEL innovations simply do not make it beyond the confines of their institution, let alone reach widespread scalable take-up (Jones and Stewart, 1999). While there are many reasons why this is so, we postulate that at least part of the problem is due to an inherent methodological weakness in the way TEL artefacts are developed. To address this problem we developed IDR, a participatory methodology for interdisciplinary techno-pedagogical design (Winters and Mor, 2008). Using the IDR methodology requires participants to step through an *identification – development – refinement* process. Each step is designed to facilitate the development of design patterns (Alexander, Silverstein and Ishikawa, 1977). The patterns are a means by which to capture and share design knowledge between participants working in interdisciplinary teams. The first stage of IDR is to identify potential patterns through the use of typologies (i.e. overview maps of a particular domain) and case studies. The next stage is to develop a set of patterns based on designs evidenced by the case studies. Once this initial set has been chosen, the third stage is to improve the patterns through collaborative discussion and reworking. The aim of working through the steps, i.e. through the *pattern development process*, is to integrate expertise across disciplines such as pedagogy, computer science, interface design and psychology. IDR was developed and used by a pan-European project team across six countries and applied with the wider community through our workshop model (Mor and Winters, 2008), run and iteratively refined at six events, involving over 200 participants. This resulted in the development of over 150 patterns (<http://lp.noe-kaleidoscope.org/outcomes/patterns/map/>). However, in reviewing the implementation of IDR, enabling participants to engage in the process was found to be equally as important as generating the design patterns as outputs. One implication of this is that the means by which participants develop the skills to write and develop patterns is critical. Only when participants understand this process can design patterns be considered to be realising their intended aim as sharable abstractions of methods for solving design

problems.

Design patterns draw their power from their mediating position between practice and theory. Patterns are often derived from concrete problems and solutions (see Section 1.1), yet phrased in a manner that affords generalisation and application in a broad set of contexts. This feature of patterns raises the question: how do we engage an interdisciplinary community of experts in the process of eliciting potent abstractions from their practical experiences? There are two main aspects to supporting this engagement. The first is to determine a coherent social configuration – a description of the ways in which collaboration occurs between members of the development team. As a pan-European team developing a pattern language, we used what we termed a *distributed development network* (d<sup>2</sup>n) social configuration. d<sup>2</sup>n is extensively described by Winters, Mor and Pratt (in press). The main characteristic of this configuration is that distributed groups develop a pattern language to share their expertise, sometimes in collaborative long-term projects, sometimes in ad-hoc exchanges.

The second is to support the abstraction process across the stages of the IDR methodology mentioned above. This support is comprised of two parts: (i) providing an appropriate way for participants to describe their practice; (ii) a means for comparing descriptions in order to elicit commonalities and derive generalisable forms. This is particularly important because the ability to abstract generalisable forms from specific instances is at the heart of the pattern elicitation, and by extension pattern development.

Furthermore, because we want to support interdisciplinary practice, we make it a requirement that the means to describe practice is easily useable by *all* participants: the approach taken must not be weighted towards any particular domain. In this way, the expertise of anyone involved in the production or use of a technology can be leveraged, supporting the aim of techno-pedagogical design.

We chose case studies as the means for participants to detail their practices. While often thought of as not supporting the production of generalisable forms, we detail related

research in support of our argument that *case study generalisation is particularly suited to developing design patterns*. This argument is fully detailed in Section 2.2 but for the moment we will say that case studies provide an interesting midpoint between very context-specific approaches and those where abstraction is absolutely paramount, for example the generation of UML diagrams. The challenge is to make case studies a part of the pattern development process in such a way that they do not veer too close to ‘context-specificity’ and thus undermine the abstraction process.

We believe that by using a clear methodology for pattern development – in our case the IDR methodology, and an appropriate social configuration – in our case, d<sup>2</sup>n, coupled with a structured process of generalisation, critiques regarding the informality of design patterns can be addressed. Our approach in this paper views the intrinsically empirical nature of patterns as a strength, seeking to structure it using appropriate levels of abstraction.

### ***1.1.Related research: methods of pattern elicitation***

Pattern elicitation is a key part of pattern development. However, it is a more complex process than it might first seem. What does it mean to identify patterns in practice?

Alexander initially provided the answer to this question. He used a phenomenological approach to identify places that have “the quality without a name”. According to Dearden and Finlay (2006), he was:

trying to identify the distinguishing characteristics that account for the success of the selected design solution [seeking] to identify key “invariants” that are common to all good solutions to that design problem and not present in poor solutions.

The key point is to identify high quality “invariants”. But how? Baggetun, Rusman and Poggi (2004) identify two main approaches to this problem from the design patterns literature: *deductive* pattern elicitation and *inductive* pattern elicitation:

- Deductive processes begin with general views and move toward specific ones.
- Inductive processes begin with specific views and move toward general ones.

Inductive methods for eliciting patterns include:

- Ad-hoc discussion. This has been used in the field of computer gaming. Patterns are “taken verbatim from a game post-mortem” (Kreimeier, 2002). Such discussion can be based on elements of game design (for example, objects should act in a predictable manner), or game narratives.
- Structured analysis and play testing of games (Bjork and Holopainen, 2005): Structured analysis is premised on “understanding what patterns exist in game design” (ibid). Play testing is used to identify patterns that appear in game play, as distinct from the actual game.
- Multi-disciplinary description and validation: Hernández Leo, Asensio-Pérez and Dimitriadis (2004) developed collaborative learning patterns, “identified and described by collaborative learning practitioners, and validated by pedagogy experts using a formalism based on natural language”
- A systematic pattern development cycle (Retalis, Georgiakakis and Dimitriadis, 2006): This begins from the premise that patterns “form structured sets” (ibid) and the aim is to “reverse-engineering the systems that embed good design” (ibid). Once the patterns have been identified, they are interrelated to form a pattern language (van Welie and van der Verr, 2003). The power of the approach advocated by Retalis, Georgiakakis and Dimitriadis is that it has a foundation in systems design, and therefore a concrete starting point. The process is further supported by existing techniques (in their case from collaborative learning) to support authentic scenario design.
- Detecting patterns in IMS Learning Designs of existing courses (Brouns et al, 2005). Their approach is to apply natural language techniques, such as Latent Semantic Analysis (LSA) to IMS LD elements. Units of learning are used as input and a ‘pattern-matching’ technique determines structural similarities. The output is a skeleton structure of common elements across the input dataset.

Deductive methods for eliciting patterns include:

- Workshops: In this popular approach, participants present patterns they are working on and feedback is provided. This approach has been particularly useful

in the software development community, where writers' workshops are held at the annual Pattern Languages of Programs (PLoP) conference. The aim is to provide feedback on submitted patterns in the form of face-to-face peer review.

- **Shepherding** (Harrison, 1999): The process by which design patterns are commented on by other pattern developers, usually prior to writers' workshops.
- **Ontologies as formal specifications of shared semantics** (de Moor, De Leenheer and Meersman, 2006): The process begins with pre-existing organisational ontologies. **Pattern representation and matching techniques** are based on "ontology alignment and meaning negotiation" (ibid). This is further explored in Section 4.
- **Schuler (2001)** led a project aimed at collating a pattern language for civic participation. Patterns were solicited by an open call circulated to a range of mailing lists. 170 patterns were received from many continents, and reviewed by a committee of 34 members. 64 were selected for presentation and further refinement. The final choice and editing of patterns was undertaken by a programme board, and did not follow a strict methodology.

Each of these approaches is suited to their domain of use. For TEL, we wish to determine how the pattern development process could work. More specifically, how can patterns be developed focused on the interdisciplinary design of TEL software and associated pedagogic activities? In our case, part of this process involved capturing practitioner experiences through cases. As distinct from the analysis of systems, these cases were derived from *artefact design and use*. Therefore, our approach to pattern development requires TEL researchers and practitioners to undertake reflective design. Each participant in the (interdisciplinary) design process is gaining an increased understand, not only of patterns, but also of the different perspectives of TEL. In this sense, the process of design pattern development is a learning process, one focused on the practice of abstraction and generalisation.

## 2. Case studies as the basis for pattern elicitation

### 2.1. Motivation

Participants new to the pattern development process often encounter the ‘cold start problem’ of identifying potential patterns in data (Retalis, Georgiakakis and Dimitriadis, 2006). Given particular descriptions of practices or artefacts, how can participants determine what can be generalised and thus could potentially become a pattern? To bootstrap the process, it is necessary for participants to have an easy and accessible way to describe their experiences/contexts/artefacts. It is also essential that the capture format provide ‘ways-in’ for other participants. Such a description, often in narrative form, provides a mechanism for communication between participants and a framing structure for working toward potential seed patterns. The aim is for participants to begin by producing “operational patterns”, which are usually closely related to their practice. However, at this stage, it is important to strike a balance between developing operational patterns and moving too quickly towards developing those at a higher level of abstraction.

The format we chose for participants’ descriptions was the case study. Flyvbjerg (2006) motivates the role of case studies in learning, citing phenomenological studies to support the view that *context-dependent* knowledge (of the type produced by cases) is fundamental to people becoming experts. The “nuanced view of reality” provided by undertaking a case study is required:

Common to all experts, however, is that they operate on the basis of intimate knowledge of several thousand concrete cases in their areas of expertise. Context-dependent knowledge and experience are at the very heart of expert activity. Such knowledge and expertise also lie at the center of the case study as a research and teaching method or to put it more generally still, as a method of learning. Phenomenological studies of the learning process therefore emphasize the importance of this and similar methods: It is only because of experience with cases that one can at all move from being a beginner to being an expert.

The context-dependent knowledge provided by case studies is a good starting point for



the development of design patterns. Tripp (1985) notes that an increasing number of teachers are undertaking naturalistic research, using case studies to describe their practice as part of an action research methodology. Yin (1994) posits that case studies work well for describing interventions (in our case TEL artefacts) and the settings in which they occur. Stake (1978) states that “understandings of human affairs are for the most part attained and amended through personal experience” and that case studies capture this in a manner which is both “down-to-earth and attention holding”.

Schank and Abelson (1977) argue that stories about one's experiences, and the experiences of others, are the fundamental constituents of human memory, knowledge, and social communication. They call for a shift towards a functional view of knowledge, as Schank (1995) explains: “intelligence is really about understanding what has happened well enough to be able to predict when it may happen again” (p. 1). Such knowledge is constructed by indexing narratives of self and others' experiences, and mapping them to structures already in memory. While Schank and Abelson come from an AI perspective, their theory is supported by recent psychological studies. Atance and O'Neill (2005) define episodic future thinking as the ability to project oneself into the future to pre-experience an event. This, they claim, is a uniquely human phenomenon which precedes semantic future thinking (Atance and Meltzoff, 2005), and provides the developmental basis for skills such as planning and causal reasoning.

Our conception of case studies sees them as *structured narratives* of expert practice. Bruner (1986; 1990; 1991; 1996) identified narrative as the predominant vernacular form of representing and communicating meaning. Humans use narrative as a means of organizing their experiences and making sense of them. A narrative is always contextualized. It habitually begins with an exposition, which lays out the context: time, location, props and characters. Bruner (1990) discusses scripts as the indigenous form of encoding knowledge. A script is a recipe for solving problems. It includes the context in which it is applicable, the sequence of operations to carry out, and the expected implications. Narrative, as a cognitive and communicational construct, has a central role in constructing and sharing scripts. A narrative is grounded in a context, describes a

sequence of events, and implies expected outcomes. It serves as the prototype after which the script is moulded.

Recent advances in neural psychology ground the development of narratives in new understandings of the brain's inner working (Mar et al, 2006; Mar, 2004; Holyoak & Krogen, 1995; Young & Saver, 2001; Addis et al, 2004; Mason, 2004). Other findings point to a strong link between narrative comprehension and theory-of-mind processing (Mar, 2004), suggesting that the cognitive modelling of the storyteller and the protagonists is a critical constituent in understanding a story. Taken together, a pattern of learning emerges from these findings. During narrative comprehension, we need to select and then connect the key events of the story. The selection of events utilizes contextual and temporal cues, while connecting them relies on causal inferences. These causal inferences are the genesis of a shift towards a semantic representation of the narrative. In the process, episodic memories are invoked, where they bear resemblance to those in the current narrative. At the same time, causal relationships are retrieved from long-term memory and tested for adequacy. Thus, the perceived narrative is understood by weaving it into a semantic network of past concepts and experiences. As a result, it loses its specificity and becomes itself part of this network, ready to be used as a script to predict future events and guide actions, or as a template to assist in the understanding of future narratives. The neural evidence shows that similar mechanisms are invoked in narrative comprehension and construction.

With respect to the development of design patterns, processes of abstraction and generalisation are critical. Therefore, an understanding of the relationship between narrative case studies and generalisation is required.

## ***2.2. Case studies, abstraction and generalisation***

Case studies have long been seen as “immune to generalisation” (Tripp, 1985), presenting a tension for anyone considering their use. On the one hand, as a starting point for pattern elicitation, developing cases is both familiar and practical for practitioners. They provide

a broad scope for participants to detail the successes of their everyday experience and practice and are a good focal point for discussion. On the other, each case study is a subjective interpretation of the practitioner's own work. This is a significant issue because design patterns by their very nature describe “the core of the solution to [a] problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” (Alexander et al, 1977, p.x). They involve a process of abstraction in order to produce generalisable solutions to problems. Therefore, the ability of participants *produce generalisations* is key to the development of design patterns. However, this is not a trivial problem. It is well known that expressing generality and articulating structure are notoriously elusive for most people. As Radford (2000) demonstrates, learners talk metaphorically about the general through the particular. Noss, Healy and Hoyles (1997) argue that abstractions are predominantly situated: bound to a specific context and expressed in a language derived from it. Indeed, in Winters and Mor (2008) we argued that context is a critical element of design patterns. The difficulty arises in the shift from case descriptions to abstract patterns. Here, the context, the problem and the solution need to be abstracted and generalised, while maintaining enough substance to be readily applicable to new concrete problems.

In relation to pattern elicitation then, a question arises as to the ways in which case studies might offer a ‘stepping stone’ to generalisation. Indeed, this issue is controversial. Abercrombie, Hill and Turner (1984) portray the conventional wisdom when they suggest, “a case study cannot provide reliable information about the broader class”. Campbell (1975) suggests that case studies “have such a total absence of control as to be of almost no scientific value”. However, he undertook “an extreme oscillation away from my earlier dogmatic disparagement of case studies” (Campbell 1975, p. 179, cited in Flyvbjerg, 2006). By contrast other researchers (e.g. Flyvbjerg, 2006; Mjoset, 2006; Tripp, 1985) strongly promote the view that generalisation from case studies *is possible*. Their approach is subtler, detailing the nature of the educational generalisations and their relationship to ‘low-level’ theory.

As discussed in Section 2.1, we see narrative as a key element in the process of *situated*

*abstraction* (Noss & Hoyles, 1996; Noss et al, 1997), forming a path from experience to knowledge. Recent findings in neuropsychology (Mar, 2004) suggest a strong link between experience, narrative and abstraction. One of the fundamental mechanisms by which we form generalised knowledge is by comprehending or constructing narratives. We do so by linking the events in a narrative to our memories from events we experienced.

### **2.3.Naturalistic generalisation**

There are many forms of generalisation. In this paper, we concentrate on naturalistic generalisation, the process of “recognizing the *similarities* (italics added) of objects and issues in and out of context and by sensing the natural covariations of happenings” (Stake, 1978). Naturalistic generalisations are significant in relation to design patterns because they are derived from people’s experience. However, Stake (1978) notes that

These generalizations may become verbalized, passing of course from tacit knowledge to propositional; but they have not yet passed the empirical and logical tests that characterize formal (scholarly, scientific) generalizations.

They are useful then where we wish to investigate similarities between particular cases, rather than generalising to entire populations. It is the similarities that are the starting point for naturalistic generalisation. Tripp (1985) points out that there are a number of different types of similarity statements: literal, metaphorical and analogy. We focus on literal similarity, i.e. those between two cases that share a number of “high-salient attributes” (ibid). For this naturalistic generalisation to succeed, a method of documenting the high-salient attributes, i.e. the common features across cases, is required. These can be differentiated between comprehensiveness – features unique to a case, and comparability – features common to a group of cases (ibid). Design patterns need to capture comparable features.

### 3. Our approach: a structured process of generalisation

The approach described in this paper emerged mainly from the work of two projects: WebLabs (<http://www.lkl.ac.uk/kscope/weblabs/>) and Learning Patterns (<http://lp.noe-kaleidoscope.org/>). In WebLabs we successfully designed tools and activities for mathematical learning, while in Learning Patterns we tried to identify the critical elements of design, which contributed to successful practice in WebLabs and in other projects involving our European partners. Our starting point in the Learning Patterns project was both promising and challenging: on the one hand, we had a team of researchers highly experienced in designing rich learning situations. On the other hand, this team was diverse in background and distributed geographically in the manner described by the  $d^2n$  social configuration (Winters, Mor and Pratt, in press). How could they engage in pattern development via case study generalisation?

Our first undertaking was to establish a common vocabulary. This was manifested in the form of a set of domain-specific typologies. Each typology was essentially a hierarchical glossary of key concepts in a domain of expertise. These typologies consciously evaded the hard questions of semantic relations, characteristic of ontology design: they were aimed at facilitating human communication and thus could enjoy the benefit of assuming tacit knowledge. The typologies acted as a semantic starting point for discussion around interdisciplinary design and capture particular aspects of TEL design knowledge.

The purpose of case study development by participants is multifold: i) to provide concrete examples of practice within disciplines; ii) to map practices and content detailed in the case study to the set typologies; iii) for the team to identify linking points between disciplines; and iv) to provide the starting point for pattern development.

Participants begin creating case studies by providing a name and a short summary. They are then directed to an editable online document based on the case study template (Figure 1). This template prompts them to provide the context (with links to the typology), aims, details, outcomes and references. The main bulk of the case study is expected to reside in the details section, which is a free-form narrative. Participants were encouraged to

include graphical materials, such as screen shots and diagrams. Our templates are all ‘soft templates’: they offer a structure, but do not impose it. The participant has full artistic licence to describe her case study in any way she finds most fitting.

demo			[Edit Info]
Created	Yishay Mor, 03 April, 2008	Modified	03 April, 2008
Summary			[Edit Page]
			[Publish]

#### Context

*Provide details regarding the environment from which this case study is derived. Who was behind the game development / deployment? Who was the target audience? What is the broad domain of knowledge / expertise addressed? Use the table below as appropriate, or enter your own free-text description.*

mathematical content	content domain: ; target audience: ; Skill Domain: ;
Learning and Instruction	Mathematical content: T2; Didactical functionalities: T2; Metaphor: T2; Rationale: T2; T2; T2;
Educational Context	production: T3; cultural: T3; teacher support: T3; intervention: T3; T3; T3;
Games	Game as activity: T4; games as genres: T4; game as media: T4; T4; T4; T4; T4;
Interface and interaction	user interface: T5; accessibility: T5; user setting: T5; T5; T5; T5; T5;
Software Design	Requirements: T6; Development Methodology: T6; Customisation: T6; T6; T6; T6; T6;

#### Aims

*What were the intentions of the parties involved? What were the expected outcomes? What are the measures of success?*

#### Details

*Describe the game itself, its process of development and / or deployment, focusing on the issues which you find most interesting. Write in free narrative form, with a clear reference to your personal experiences and observations. Include visual materials if possible and link to on-line resources.*

#### Outcomes

*What were the positive and less desirable results of this experience? How do they map to the original aims? What unexpected outcomes (good or bad) have you noted? What are the key factors contributing to the outcomes?*

*Figure 1: the case study template provided to participants*

The next phase aimed to identify the substance from which our patterns would be constructed: incidents of powerful, generalisable design carefully selected from the team members' experience. Here, we encountered the “cold-start” problem discussed above (Retalis, Georgiakakis & Dimitriadis, 2006). Many of the team members found the fundamental constructs of design patterns challenging and confusing. As TEL researchers, we are trained in meticulous reporting and analysis, and in theoretical abstractions. Patterns aim to fill the gap between these extremes – a gap we are not confident in addressing. To ease the process, we decided to begin by sharing case studies, in narrative form, and discussing them through project meetings and on-line forums. Team members were instructed to relax their scientific accuracy, and focus on the

narrative: the context of the experience they related, the central problem addressed, the key steps taken to resolve it, and the perceived outcomes. In hindsight, our case studies had a phenomenological quality, highlighting the contributors' perceptions of events rather than a claim to subjective evidence, and echoing Alexander's original approach.

Unsurprisingly, the diverse backgrounds of team members resulted in a diversity of language used to describe cases, which gave rise to a seemingly insurmountable barrier to comparability and by extension, generalisability. This is where the typologies played a critical role. Having drafted their cases in their own language, team members were asked to revise them and rephrase the context and problem description by selecting the appropriate terms from the typologies. This process led to a refinement of the typologies as well as the cases. Often a team member would find that a particular typology lacked the detail needed to describe her case. This would prompt a discussion with the owner of the typology, to elaborate it and resolve the deficiency. Once we identified the viability of this process, we scaffolded it by providing a template for case studies, which prompted contributors to mould them into the common form.

Once the case studies had been mapped to the common vocabulary of the typologies, an initial set of similarities began to emerge. In some cases, a match in the context or problem description led to divergent solutions, which gave rise to an attempt to identify common fundamental components within these solutions. In other cases, analogous solutions in seemingly different contexts suggested a common class of problem at a higher level of abstraction. Such incidents generated our first seed patterns: skeletal notes that include little more than name and synopsis. Each seed pattern earmarked a triplet of context, problem and solution scheme to be elaborated later. The actual progression from scattered seed patterns to a fully-specified and networked pattern language is at the core of the IDR methodology (Winters and Mor, 2008).

Reflecting on our own process and its success, we identified a method of *structured narratisation*: abstraction of design patterns from narrative accounts of personal experience by scaffolding them with shared typologies and a common template, driven

by interdisciplinary discussion.

As our language of patterns began to take shape, we initiated a series of public workshops (Winters and Mor, 2008; Mor and Winters, 2008). The aim of these workshops was to verify both the pattern language and the methods of its development, by exposing them to a wider audience. Using the typologies and the case study template, workshop participants were led through the same process of structured narratisation. Whether in the intimate scope of the team, or the broader circle of workshop participants, the key principle was maintained: pattern development was a community endeavour, and proceeded through a communicative process.

Workshops generated new patterns, and in parallel team members refined and elaborated the patterns they had previously contributed. As with case studies, a template for scaffolding patterns evolved in due course. This template was much more detailed, including a richer context, an up-front statement of the problem, and slots for designating links and weaving patterns into a coherent language. Comparing the more detailed patterns, and mapping them to new case studies, often led to refactoring<sup>1</sup>: a common element is found in a number of patterns, and extracted as a pattern in its own right. Even though the patterns have by now been dramatically abstracted away from the case studies in which they originated, it is important to retain a link back to them. Any reader of the patterns, who has not been party to their development, will find these links indispensable to their understanding.

Both the individual authoring of cases and patterns and their collaborative refinement are supported by a set of bespoke software tools (Winters and Mor, 2008). These tools evolved in tandem with our practices, in a dynamic of social instrumental genesis.

### ***3.1.Example: the GmX trail***

During 2003-2005 we conducted an experiment called Guess-my-Robot (Mor et al, 2006)

---

<sup>1</sup> A term borrowed from object-oriented programming.



with students across Europe. This experiment involved a game in which learners posed mathematical challenges to each other in the domain of number sequences. One learner would construct a computer program which generated a number sequence. She would retain the program and publish the first few terms of the sequence on a web-based collaboration system, as a challenge to her peers. They in turn would try to uncover the program that generated the sequence, and if successful – post it as a response.

The success of this game led to the development of two new activities, in different domains: Guess-my-Garden (Cerulli, Chiocciariello and Lemut, 2007), which was a similar game in the domain of probability and Guess-my-Graph (Simpson, Hoyles and Noss, 2006), in the domain of functions and graphs.

Both Guess-my-Robot and Guess-my-Garden were captured as case studies. We described them in intricate detail, including graphics and examples from the original game sites. We identified a pattern, common to Guess my Robot and Guess my Garden, which we called *Guess-my-X* (GmX). GmX is a complex and detailed pattern (see Appendix A). We realised that it is useful to abstract and decompose it to more modular and generic elements. Going up the abstraction ladder, we noted the *Challenge Exchange* and *Build This* patterns (see Section 4.1). Reflecting on the patterns we had identified, we realized they were present in other cases. For example, *Build This* underlies the design of *Programming building blocks* (see Section 4.1). As for components, we identified the patterns *Objects to talk with*, *League chart* and *Active Worksheet* (For a full list of our patterns, see: <http://lp-noe.kaleidoscope.org/outcomes/patterns>).

## 4. Discussion

Carroll and Farroq (2007) consider “patterns as a paradigm for theory”, providing a framework for generalising case study descriptions. In our approach, generalisation was supported through the identification of salient features across comparable cases. It is clear that determining the salient features is a non-trivial problem. The typologies we developed were a starting point of this process. However, developing appropriate

typologies requires expert knowledge derived from experience. Starting from typologies that are inappropriately constructed will lead to problems when trying to determine the salient features later in the pattern development process. This is similar to the problems faced in interorganisational ontology engineering, in particular “multiple types of context dependencies need to be handled between ontological elements” (de Moor, De Leenheer, and Meersman, 2006). However, because typologies are less structured than ontologies, participants were free to augment them during the pattern development process, in line with evidence emerging from the case studies.

A key aspect of developing appropriate typologies was the representation used. We chose visual mindmaps, which proved to be both accessible and easy to edit. As detailed by Simon (1969), representation is important in order to illuminate our capacity to solve problems – in this case, to support the processes of abstraction and generalisation. This theme, which emerges from Mor and Winters (2007), is a subject of our current research.

#### ***4.1. Patterns and abstraction***

One of the critical challenges that developers of design patterns face is how to determine the *level of abstraction* when attempting to identify high-salient features. To provide a concrete example we will look at the two cases studies mentioned above: ‘Guess-my-Robot’ and ‘Programming building blocks’. Partner institutes with different expertise (the London Knowledge Lab and the Freudenthal Institute, respectively) submitted each of these example cases. In brief, ‘Guess-my-Robot’ is a challenge game, where students have to guess each other’s number sequences. ‘Programming building blocks’ concerns block programming, where students’ construction involves providing the 3D co-ordinates of the blocks. At first look these examples do not seem to have much in common: They are aimed at different domains within mathematics and are designed for different age groups. However, upon closer inspection we see that in both cases, a finished ‘product’ is provided to students (a ‘robot’ or a ‘building block’). This is the first salient feature and identifying it partially relied on mathematics teachers’ experience and expertise in understanding the types of challenges learners will engage with.

A second salient feature is the type of learning activity associated with each case. Both involve the challenge of reconstructing the code that generated the finished ‘product’, as captured with the *Build this* pattern. As noted in Section 3.1, this pattern involves a high level of abstraction: It is based on looking for a context-independent feature, i.e. one that will work when applied across different contexts, in this case different mathematics classrooms. Thus, it is higher up in our pattern language. Moreover, it exemplifies the fact that when discussing the generalisability of case studies in relation to the development of patterns, the issue of context is critical. In TEL, the nature of the relationship between patterns and their context is particularly important to teachers. Tripp (1985) provides a relevant example. He determined that it is important to differentiate between what is “context-bound” and what is not. Patterns need to capture what is true across contexts, but also which elements of the context are retained as necessary conditions for their applicability. For example, one might look at why a particular activity was engaging rather than the topic it was used for, i.e. activity design versus activity content. As an example from our work, the high-salient features of ‘Guess-my-Robot’ can be applied in a different context. This produced a similar ‘Guess-my-Garden’ case. The process is to keep the structure of the game and the underlying educational principles, but to change the mathematical *content*. Critically, the designer can only begin to think about changing the content if the educational principles are context-independent. This process was captured in the ‘Content Morph’ pattern. Tripp notes that:

Such a movement from particular to particular is generalisation, but a form of generalisation which is directly useful to teachers precisely because its location in context enables them to judge whether it is useful in their situation and how much of what kind of allowance are required

To take another example, the same is also true when applied to the interface design of TEL software. Interface design patterns gain particular power because the designs are useable across software applications – they are at the level of the interface.

What is very significant about the ‘Content Morph’ pattern is that it captures the

abstraction process. It is not a pattern derived from the salient features across cases but is a pattern about the *process of abstraction*. It details a generalisable practice for determining the context-independent aspects of cases. As such, it is a means by which the abstraction process can be bootstrapped in other projects.

The process of searching for comparative features is a learning process for any pattern developer. By having to ask questions of one's case and others, with respect to what is unique about it and what might be useful in other contexts aids thinking at a higher level. In our case, the typologies were a starting point for structuring the process of generalisation. Also key were the different visualisation viewpoints of the language. These allowed participants to easily get a feel for the overall language, drilling down to determine gaps and inconsistencies. A number of viewpoints were offered (see: <http://lp.noe-kaleidoscope.org/outcomes/patterns/map>): the primary view, or *list-view*, is a browsable, hierarchical list of current patterns in the language. When you mouse over each pattern, you are provided with a 1-line summary explaining the purpose of the pattern. This is consistent with the pattern's name. The primary list-view is augmented by a number of secondary views. The *overview* is a Freemind mindmap of the current structure of the language; the *live-view* is a java applet that combines the functionality of over- and list- views; the *index-view* produces a list sorted on a particular criterion (pattern name, submitted by, first listed, last edited, summary, category, status and rank).

#### ***4.2.Deriving patterns from cases as a means to accumulate design knowledge in TEL***

Design patterns, by their very nature, are cumulative in the sense that they synthesize knowledge. The software development community has adopted design patterns as an effective form for accumulating design knowledge. Novice programmers can create code by using readily available, rich collections of patterns. This ability draws on two characteristics of their field of practice: first, the nature of the designed artefacts is such that they lend themselves to a direct mapping to and from patterns. Indeed, many software development tools automatically generate code templates from patterns. The

second factor is the nature of the practice itself. Programmers are trained to abstract problems and implement solutions. Moving up and down, from the concrete to the symbolic, and back again becomes a second nature. Hence, the pattern formulation appeals to software engineers' intuition.

The situation in TEL research is somewhat different: the accumulation of knowledge is known to be a serious challenge. The vision document of the EC Kaleidoscope Network of Excellence highlights the need to address “the lack of accumulation and utilisation of current practice” (Kaleidoscope, 2007, p 7). In Mor and Winters (2007), we suggested that design patterns have the potential to address this problem by offering researchers a common language for expressing and connecting their findings. We wish to qualify that statement, based on our observations above.

We note that over the last decade there has been substantial work in developing patterns in TEL, and education in general (examples include Anthony, 1996; Bergin, 2000; Avgeriou et al., 2004). Yet the uptake of this work by the wider community has been disappointing. To understand this observation, and its consequences in terms of knowledge accumulation, we return to a distinction we made in Mor and Winters (2007) between the analytical (or descriptive) and normative roles of patterns, and their position in the wider context of design based research.

In a nutshell, the analytical function of patterns pertains to their function in a structured and semi-formal report of *how things are*, how good design is conceived. They are analytical in the scientific sense of the word: they decompose the object of study into its constituents and identify the relationships between them. By contrast, the normative function of patterns is directed at claiming *how things should be*. Alexander's patterns are arguably of this kind, marked by an imperative form and modest claims for empirical evidence (Alexander et al., 1977).

Both functions of patterns are valid and important; both provide a form of knowledge accumulation. The first allows us to provide richer descriptions of the artificial world by

building on others' previous contributions. The second holds a promise to feed those new understandings back into improved practice. This is where a central tenet of design-based research (DBR) comes to play. DBR is an inherently iterative methodology (Cobb et al, 2003), based on the proposition that theory and practice need to constantly co-evolve. Analysis of design should lead to ontological innovations (diSessa and Cobb, 2004), which in turn inspire better design.

The lack of widespread adoption of design patterns in TEL means that the accumulation of knowledge afforded is linear rather than cyclic: good practice is recorded and analysed, but the resulting ontological innovations do not induce novel designs. We do not enjoy the amplification of the feedback loop. Knowledge is aggregated, but not repurposed.

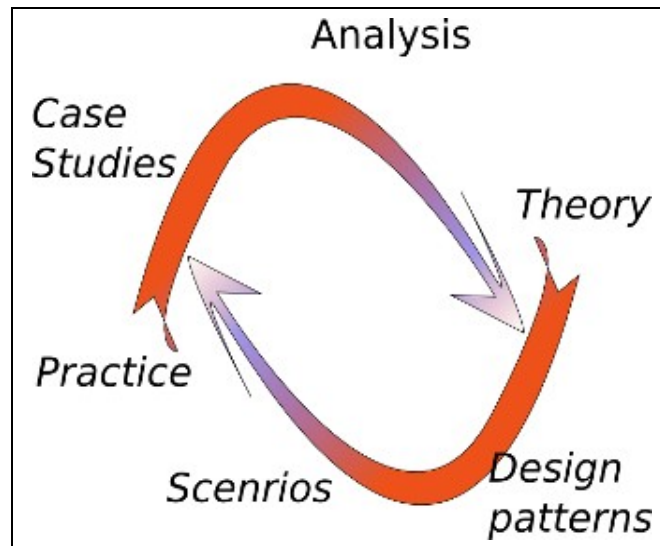
Perhaps the reason for this failure is that the cognitive habits of educational practitioners differ from those of software engineers. A teacher is required to constantly provide real-time ad-hoc solutions to unique situations. Such experience fosters skills which are quite the opposite of the abstraction-implementation dialectic mentioned above. Whether this is the root cause or not, the phenomenon we observed was quite clear: abstract design patterns were not an approachable representation of knowledge for teachers, educational designers and even educational researchers. We were left in a frustrating situation of presenting our peers with aggregated knowledge, which we were strongly convinced was relevant to their work, and seeing it politely rejected. This situation, however, changed dramatically once the patterns were linked back to the case studies. Such linkage allowed our audience to apply the familiar tools of analogy, connecting the patterns to their own experience. By extending this analogy, they could then filter their experience by the pattern, identifying the salient features it highlights in a familiar context and weaving the new knowledge it presents them into their existing conceptual web. This approach has not been adopted by the pattern language network project ([patternlanguagenetwork.org](http://patternlanguagenetwork.org)), which is applying a similar methodology in the domain of using social technology in higher education.

Even so, our patterns still need to face the ultimate test: will they be used to develop novel designs, in expanding contexts? Will these designs lead to enhanced learning? Unfortunately, most of the evidence we have here is anecdotal: through personal communication, we have heard of successful use of these patterns. But most of these have not been documented or evaluated systematically. The single notable exception we can report is the Guess-my-Die game, derived from the Guess-my-X pattern (Winters, Mor and Pratt, in press).

### ***4.3.Successes, limitations and future work***

The approach described in this paper was developed with notable success by the Learning Patterns project. The project led 6 workshops, which attracted scores of participants. These workshops brought together researchers, designers, developers and educators from diverse fields such as teacher education, medicine, mathematics and computer science. The meticulous combination of case studies and patterns allowed uninitiated practitioners to share their experiences, identify salient recurring design features and bring these into a common design language. As a result, close to 30 case studies and 150 patterns have been generated, debated and refined.

Nevertheless, the challenge of systematic evaluation is still before us. To do that, in the framework of design based research, we will need to work closely with stable practitioner communities over extended periods of time, following them through the full cycle of design-based research, deriving design patterns from their art and implementing new designs from patterns. The pattern language network is exploring this question by introducing the construct of scenarios: hypothetical case studies derived by applying new patterns to the premises of existing cases (Figure 2). Other projects, which are still in conception, aim to achieve this goal in other domains of knowledge and practice.



*Figure 2: Manifesting the full cycle of design based research with a pattern-based methodology: empirical data from design experiments is captured as case studies, these are analysed using domain-appropriate scientific methods and calibrated with theory. As an outcome, design patterns are derived and used to revisit the case studies and generate scenarios for the next round of experiments.*

The effective and lively presentation of case studies has proved to be of critical importance. In the Como project (<http://www.lkl.ac.uk/como>) we are experimenting with the use of “comic” style storyboards to support richer narratisation of scenarios and cases. Such storyboards have an advantage of inherent abstraction: the superfluous detail is removed and what is left is a step closer to pattern diagrams. We intend to develop this method further, and combine it with other techniques of digital storytelling. In particular, we are investigating media for (wiki-style) collaborative construction and discussion of such storyboards.

## 5. Conclusion

The focus of this paper was on the process of developing a pattern language, thus supporting abstraction as a form of collaborative learning within an interdisciplinary design community. We identified a method, which progressed through the co-construction of typologies, case studies and patterns. In particular, we noted the process of *structured narratisation* which is sustained by the carefully scaffolded elicitation of



patterns from case studies. We considered the potential of patterns and cases to support knowledge accumulation in TEL, and noted that this should not be taken for granted. It is ultimately contingent of sustaining the full cycle of design-based research, using patterns to allow practice to inform theory and theory to inspire practice. We see two critical challenges for our future work: first, to validate our arguments (and our patterns) by establishing this complete cycle in a rigorously monitored manner. We view this process as a way to evaluate the full potential of our language. It is an opportunity for further development of generalisations across the language and a means to determine commonalities and relationships that may not yet be evident. Second, to identify powerful dynamic visualisations of design patterns and design knowledge more generally. This will be an extension of our work on viewpoints, and a potentially productive means of supporting increased understanding of the sometimes complex relationships between patterns.

This paper emerged from an exploratory design-based study. The value and impact of the approach is demonstrated by the outputs of that study, as described in Section 4.3. At the same time, the indications we have for the validity of the approach call for a more extensive iteration, with broader scope and span. Moreover, we need to evaluate how the design knowledge captured by the patterns feeds back into practice. While we do have informal reports of workshop participants making use of the patterns in their subsequent work, we aim to extend our approach to cover a *full iterative cycle* of design, development, implementation, deployment, evaluation and redesign (see Section 4.3). Our current research is focused on deployment and evaluation, investigating the ways in which patterns are used in practice, leading to their redesign.

In developing patterns it is critical that interdisciplinarity practices are maintained. As such the process of abstraction is as much social as it is cognitive. Every effort needs to be made to provide a supportive environment for pattern generation. It is therefore incumbent on participants who deal with abstractions everyday to support those who do not. Similarly, those who develop cases that convey the richness and complexity of real-world experiences need to support those who do not. In this manner, the patterns

developed are not biased towards one experience or tradition. They are the result of the synergistic and interdisciplinary process characteristic of excellence in TEL. Moreover, if we see learning as tool dependent and good design as central to the production of tools, then patterns are central to TEL in a way that has – regrettably – been little noticed outside the patterns community. However, as proven by the publication of this special issue and a forthcoming book on TEL and design patterns (Goodyear and Retalis, in press), patterns' time has come. We can look forward to an interesting and productive future.

## 6. Acknowledgements

We would like to thank all the members of the *Learning Patterns* project team for their important contributions to this research: Efi Alexopoulou (Educational Technology Lab; University of Athens); Staffan Björk (Göteborg University and the Interactive Institute, Sweden); James Bligh (CRITE, Trinity College Dublin, Ireland); Mark Childs (Institute of Education, University of Warwick, UK); Michele Cerulli (Istituto Per Le Tecnologie Didattiche, Italy); Vincent Jonker (Freudenthal Institute, The Netherlands); Chronis Kynigos (Educational Technology Lab, University of Athens); Fionnuala O' Donnell (CRITE, Trinity College Dublin, Ireland); Dave Pratt (Institute of Education, University of London); Brendan Tangney (CRITE, Trinity College Dublin, Ireland); Monica Wijers (Freudenthal Institute, The Netherlands). The *Learning patterns for the design and deployment of mathematical games* project was supported by the Kaleidoscope Network of Excellence, under the EC FP6 Framework. Finally, we would like to thank the reviewers for their helpful comments.

## References

- Abercrombie, N., Hill, S., & Turner, B. S. (1994). Dictionary of Sociology (3rd ed.). Harmondsworth, UK: Penguin.
- Addis, D. R., McIntosh, A. R., Moscovitch, M., Crawley, A. P. & McAndrews, M. P. (2004), Characterizing spatial and temporal features of autobiographical memory

retrieval networks: a partial least squares approach, *NeuroImage* 23, 1460-1471.

Alexander, C., Silverstein, M. and Ishikawa, S. (1977) *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. New York, NY: Oxford University Press.

Anthony, D. L. (1996), Patterns for classroom education  
<http://ianchaiwriting.50megs.com/classroom-ed.html>. Last accessed 26 November 2008

Atance, C. M. & Meltzoff, A. N. (2005), My future self: Young children's ability to anticipate and explain future states, *Cognitive Development* 20(3), 341-361.

Atance, C. M. & O'Neill, D. K. (2005), The emergence of episodic future thinking in humans', *Learning and Motivation* 36, 126-144.

Avgeriou, P., Vogiatzis, D., Tzanavari, A. & Retalis, S. (2004), Towards a Pattern Language for Adaptive Web-based Educational Systems, *Advanced Technology for Learning Journal* 1, 202--209.

Baggetun, R., Rusman, E., & Poggi, C. (2004). Design patterns for collaborative learning: From practice to theory and back. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of the World Conference on Educational multimedia, hypermedia and telecommunications*, AACE, Lugano, Switzerland p. 2493 – 2498.

Bergin, J. (2000) *Fourteen Pedagogical Patterns*. Paper presented at the Fifth European Conference on Pattern Languages of Programs, Irsee, Germany. Retrieved June 11, 2006 from: <http://csis.pace.edu/~bergin/PedPat1.3.html>.

Bjork, S. and Holopainen, J. (2004) *Patterns in Game Design*. Massachusetts: Charles River Media

Brouns, F., Koper, R., Manderveld, J., van Bruggen, J., Sloep, P., van Rosmalen, P., Tattersall, C. & Vogten, H. (2005). A first exploration of an inductive analysis approach for detecting learning design patterns, *Journal of Interactive Media in Education* (3).  
<http://jime.open.ac.uk/2005/03>

Bruner, J. (1996), *The Culture of Education*, Harvard University Press, Cambridge, Massachusetts.

Bruner, J. (1991), The Narrative Construction of Reality, *Critical Inquiry* 18.

Bruner, J. (1990), *Acts of Meaning : Four Lectures on Mind and Culture (Jerusalem-Harvard Lectures)*, Harvard University Press, Cambridge, Massachusetts.

Bruner, J. (1986), *Actual Minds, Possible Worlds (The Jerusalem-Harvard Lectures)*, Harvard University Press, Cambridge, Massachusetts.

Campbell, D. T. (1975). Degrees of freedom and the case study. *Comparative Political Studies*, 8(1), 178-191.

Carroll and Farroq (2007) Patterns as a paradigm for theory in community-based learning, *Computer-Supported Collaborative Learning* 2:41–59

Cerulli, M. Chiocciariello, A. & Lemut, E. (2007) A microworld to implant a germ of probability, In 5th CERME Conference - Congress of European Society for Research in Mathematics Education, Larnaca, Cyprus.

Cobb, P.; Confrey, J.; Disessa, A.; Lehrer, R. & Schauble, L. (2003), 'Design Experiments in Educational Research', *Educational Researcher* 32(1), 9--13.

de Moor, A., De Leenheer, P. and Meersman, M. (2006) DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering, 14th

International Conference on Conceptual Structures, Springer-Verlag, Volume 4068, Aalborg, Denmark, p.189-203

diSessa, A. A. and Cobb, P. (2004) Ontological Innovation and the Role of Theory in Design Experiments. *Journal of the Learning Sciences*, 13 (1), 77–103.

Dearden, A. and Finlay, J. (2006) Pattern Languages in HCI: A Critical Review, *Human-Computer Interaction*, 21, 49-102.

Flyvbjerg, B. (2006) Five Misunderstandings About Case-Study Research, *Qualitative Inquiry*, 12(2), 219-245

Goodyear, P. & Retalis, S. (in press) Technology-enhanced learning: design patterns and pattern languages. Rotterdam: SensePublishers.

Harrison, N. (1999) The Language of Shepherds: A Pattern Language for Shepherding, in *Proceedings of the 6th Annual Conference on the Pattern Languages of Programs*, pp. 15-18.

Hernández-Leo, D., Asensio-Pérez, J.I. & Dimitriadis, Y. (2004) IMS Learning Design Support for the Formalization of Collaborative Learning Patterns, *Proceedings of the The 4th International Conference on Advanced Learning Technologies*, 350-354, Joensuu, Finland

Holyoak, K. J. & Kroger, J. K. (1995), Forms of reasoning: Insight into prefrontal functions?, *Annals of the New York Academy of Sciences* 769(1), 253-264.

Jones, D., and Stewart, S. (1999) The case for patterns in online learning In: P. De Bar and J. Legget (Eds.), *Proceedings of Webnet'99 Conference, Honolulu Hawaii*, 24-30. Association for the Advancement of Computing in Education.

Kaleidoscope Vision Team (2007), The Kaleidoscope Scientific Vision for Research in Technology Enhanced Learning. [http://telearn.noe-kaleidoscope.org/open-archive/browse?resource=316\\_v1](http://telearn.noe-kaleidoscope.org/open-archive/browse?resource=316_v1)

Kreimeier, B. (2002). The case for game design patterns. Available from [http://www.gamasutra.com/features/20020313/kreimeier\\_03.htm](http://www.gamasutra.com/features/20020313/kreimeier_03.htm)

Mar, R. A. (2004), The neuropsychology of narrative: story comprehension, story production and their interrelation (review), *Neuropsychologia* 42(10), 1414-1434.

Mar, R. A., Oatley, K., Hirsh, J., dela Paz, J. & Peterson, J. B. (2006), Bookworms versus nerds: Exposure to fiction versus non-fiction, divergent associations with social ability, and the simulation of fictional social worlds, *Journal of Research in Personality* 40, 694-712.

Mason, R. A. and Just, M. A. (2004), How the Brain Processes Causal Inferences in Text. A Theoretical Account of Generation and Integration Component Processes Utilizing Both Cerebral Hemispheres, *Psychological Science* 15(1), 1-7.

Mjøset, L. (2006) A Case Study of a Case Study: Strategies of Generalization and Specification in the Study of Israel as a Single Case, *International Sociology*, 21, 735

Mor, Y. (forthcoming), Guess my X and other patterns for teaching and learning mathematics, In *Proceedings of EuroPLoP 2008*.

Mor, Y. and Winters, N. (2007) Design Approaches in Technology Enhanced Learning, *Interactive Learning Environments*, 15(1), p. 61-75.

Mor, Y. and Winters, N. (2008) Participatory design in open education: a workshop model for developing a pattern language, *Journal of Interactive Media in Education*, <http://jime.open.ac.uk/2008/13/>

Mor, Y. Noss, R. Hoyles, C. Kahn, K. and Simpson, G. (2006) Designing to see and share structure in number sequences. *International Journal for Technology in Mathematics Education* 13(2):65-78

Noss, R. Healy, L. & Hoyles, C. (1997) The Construction of Mathematical Meanings: Connecting the Visual with the Symbolic, *Educational Studies in Mathematics*, 33(2), 203-233

Noss, R. and Hoyles, C. (1996) *Windows on Mathematical Meanings: Learning Cultures and Computers*. Dordrecht: Kluwer.

Retalis, S. Georgiakakis, P. & Dimitriadis, Y. (2006), Eliciting design patterns for e-learning systems, *Computer Science Education* 16(2), 105--118.

Radford, L. (2000) *Signs and meanings in students' emergent algebraic thinking: a semiotic analysis* Educational Studies in Mathematics 42(3):237—268

Noss, R. Healy, L. and Hoyles, C. (1997) *The Construction of Mathematical Meanings: Connecting the Visual with the Symbolic*. Educational Studies in Mathematics 33(2):203--233

Schank, R. C. (1995), *Tell Me a Story: Narrative and Intelligence*, Northwestern University Press.

Schank, R. & Abelson, R. (1977), *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Schuler, D. (2002), A Pattern Language for Living Communication, In *Proceedings of the Conference on Participatory Design*, Malmö, Sweden, June 23-25

- Simon, H. A. (1969) *The Sciences of the Artificial*, Cambridge, MA: The MIT Press
- Stake, R.E. (1978) The Case Study Method in Social Inquiry, *Educational Researcher*, 7 (2) 5-8.
- Simpson, G. Hoyles, C. and Noss, R (2006) *Exploring the mathematics of motion through construction and collaboration*. Journal of Computer Assisted Learning, 22(2):114—136
- Tripp, D.H. (1985) Case Study Generalisation: An Agenda for Action, *British Educational Research Journal*, 11(1), p. 33-43.
- van Welie, M., & van der Veer, G.C. (2003). *Pattern languages in interaction design: Structure and organization*. In G.W.M. Rauterberg, M. Menozzi & J. Wesson (Eds.), Proceedings of Interact, p. 527 – 534. Amsterdam: IOS Press.
- Winters, N. & Mor, Y. (2008). IDR: A participatory methodology for interdisciplinary design in technology enhanced learning, *Computers & Education*, 50(8), p. 579-600.
- Winters, N., Mor, Y. & Pratt, D. (in press). The distributed developmental network - d<sup>2</sup>n: a social configuration to support design pattern generation. In Goodyear, P. & Retalis, S. (Eds) Technology-enhanced learning: design patterns and pattern languages. Rotterdam: SensePublishers.
- Yin, R. K. (2003). *Case study research: Design and methods*. Thousand Oaks, CA: Sage.
- Young, K. & Saver, J. (2001), 'The Neurology of Narrative', *SubStance* 30(1&2), 72--84.

## **Appendix A: the Guess my X pattern**

*Following is a summary of the Guess my X pattern. The full pattern is available at:*



<http://patternlanguagenetwork.myxwiki.org/xwiki/bin/view/Patterns/GuessMyX>. An edited version of it will appear in (Mor, forthcoming).

Sustaining a mathematical discussion is vital to the establishment of socio-mathematical norms (Yackel & Cobb, 1995) and to the collaborative construction of knowledge in the community. This goal is especially difficult to achieve in geographically distributed communities. This pattern addresses this by a CHALLENGE EXCHANGE game of BUILD THIS puzzles.

## **Context**

A teacher wants to design a game for learning concepts, methods and meta-cognitive skills in a particular mathematical domain. This game should use a combination of available technologies.

## **The problem**

Learning mathematics is fundamentally learning to be a mathematician. It requires the learner to internalize a range of mathematical skills as regular habits: computation, analysis, conjecturing and hypothesis testing, argumentation and proof. For this to happen, the learner needs to take ownership of a meaningful mathematical inquiry, and engage in activities of problem solving and discussion. Games provide a natural setting for the kind of “flow” needed, but how do we ensure that the focus of this flow is mathematical activity and discourse?

## **Solution**

Guess my X is a pattern of game structure, which can be adapted to a wide range of mathematical topics. At its core is a CHALLENGE EXCHANGE of BUILD THIS puzzles, using MATHEMATICAL GAME PIECES as OBJECTS TO TALK WITH in a NARRATIVE SPACE, using a LEAGUE CHART to orchestrate SUSTAINED SOCIAL INTERACTION. A scheme of the pattern is presented in Figure 3.

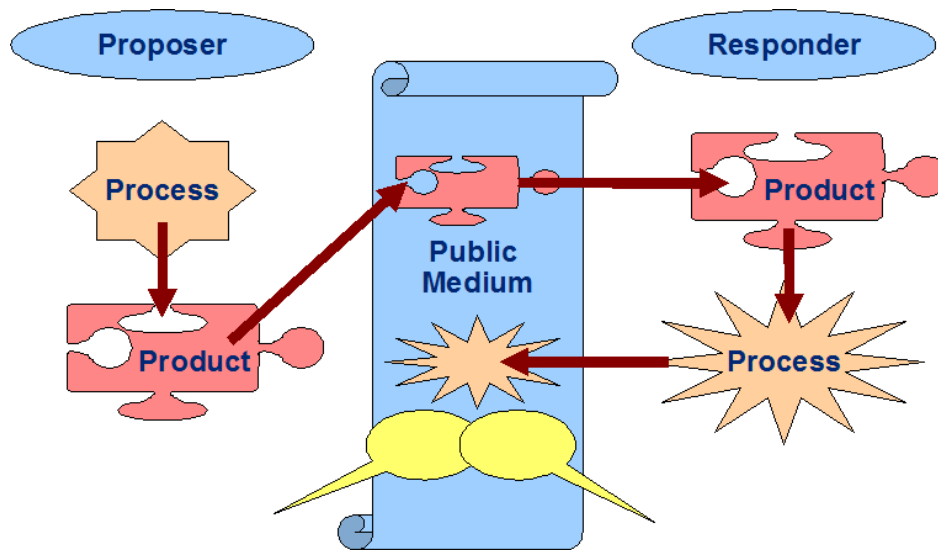


Figure 3: Schematic diagram of GmX

GmX involves players in two roles, *proposers* and *responders*, and a facilitator. An implementation of the game would specify a domain of mathematics and rules for constructing processes in that domain. A proposer sets a challenge, in the form of a mathematical object which she constructed. The explicit rules of the game define the nature of the process by which this object can be created, but not its details. The proposer would construct such a process, and capture its product. She then saves the process model in a private space and publishes the product as a challenge. Responders then need to “reverse engineer” the process from the product. If they succeed, they publish their version as a response to the challenge. The proposer then needs to confirm the responder’s solution or provide evidence for the contrary.

The game is not a SUGAR-COATING to disguise the mathematics: it is a game with MATHEMATICAL GAME-PIECES. The rules of the game are intentionally left vague. This requires students to negotiate what constitutes a correct answer, and in doing so collaboratively refine the underlying mathematical concepts. These negotiations can lead to discussions of issues such as proof, equivalence and formal descriptions. The quality and extent of these discussions depends on the scaffolding and provocations provided by the teacher, but a necessary condition for them to emerge is that the medium of the game provides a NARRATIVE SPACE, where the MATHEMATICAL GAME-PIECES of the game can become OBJECTS TO TALK WITH.